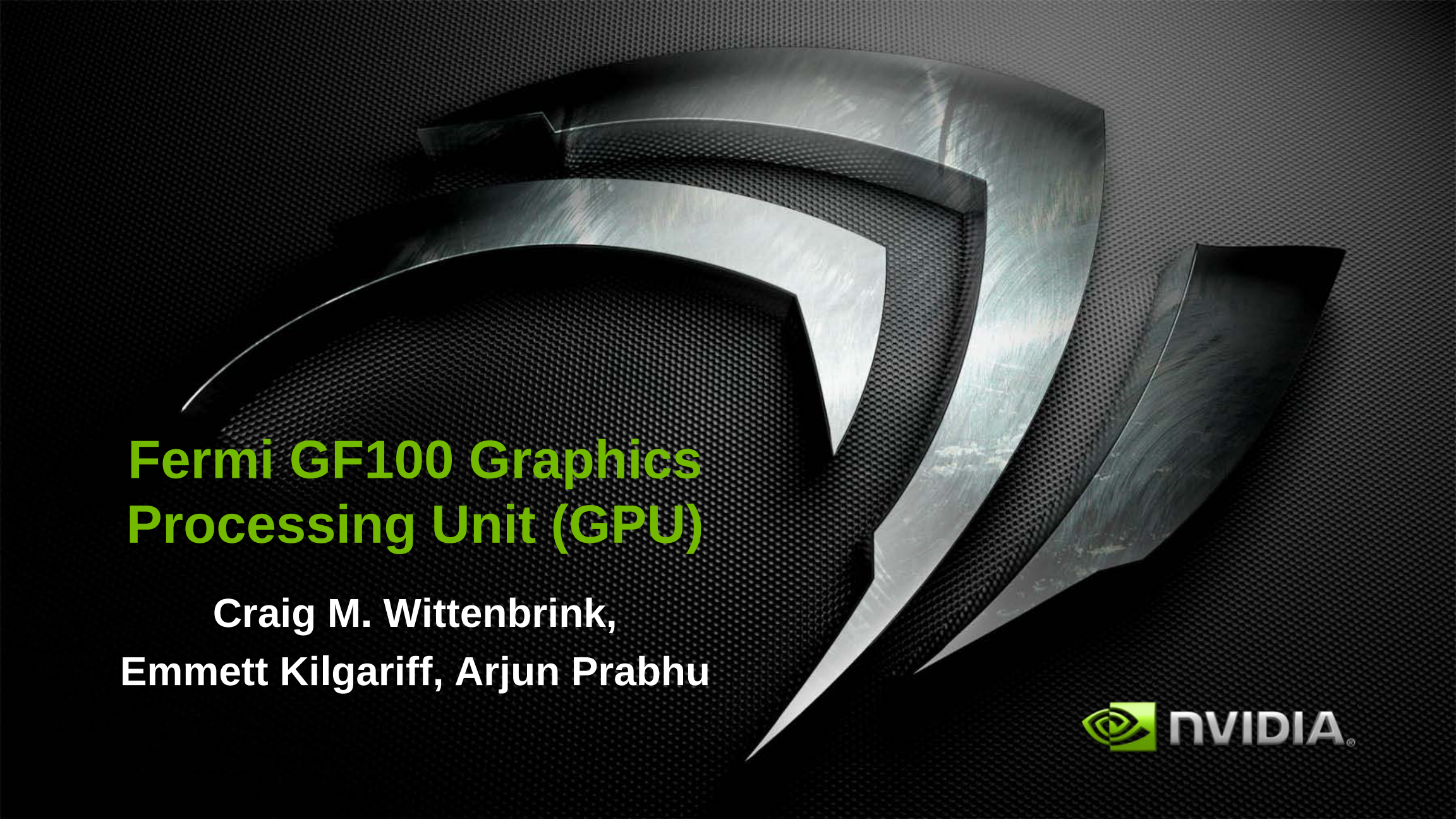


Fermi GF100 Graphics Processing Unit (GPU)

Craig M. Wittenbrink,
Emmett Kilgariff, Arjun Prabhu

NVIDIA

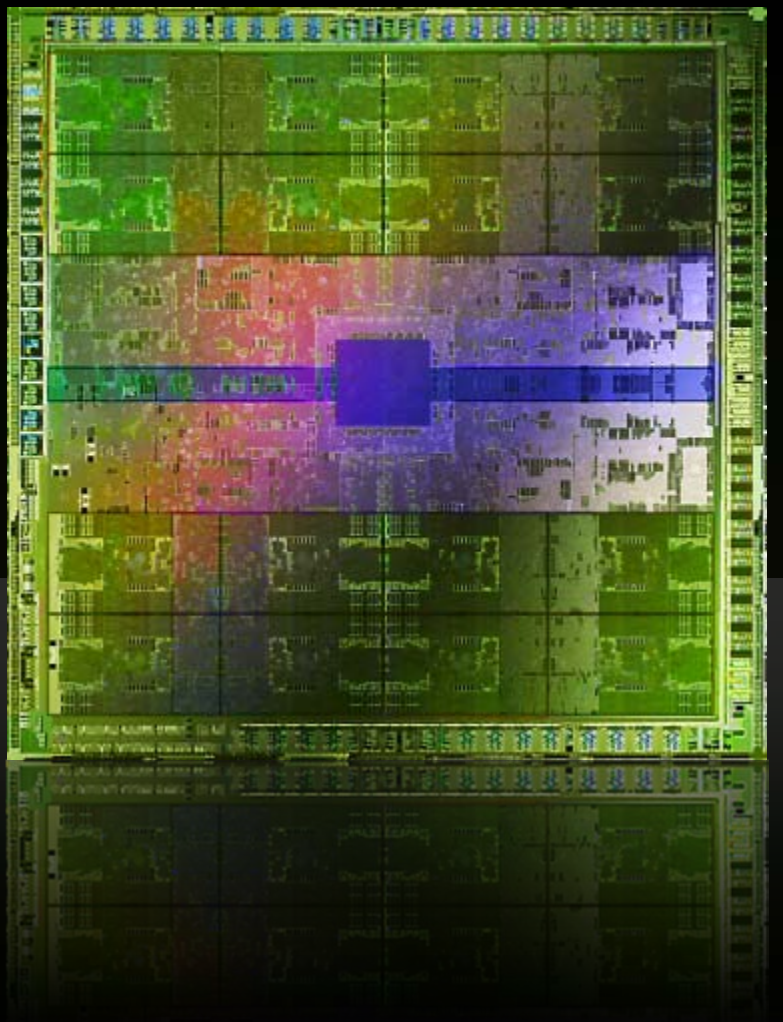
Acknowledgements

- We would like to thank Jonah Alben, John Robinson, Mark Daly, Henry Moreton, and the entire Fermi GF100 Development team for success of this project

Outline

- GF100 Specifications
- GF100 Architecture
- Tessellation
- Physics
- Computational Graphics
- Demos

“GF100”



GF100 Specifications

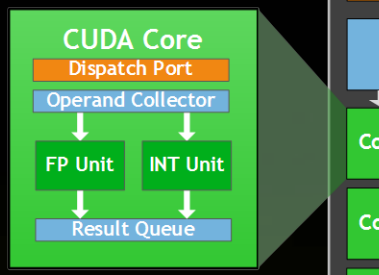
- 3 Billion Transistors in 40 nm process (TSMC)
- Up to 512 CUDA / unified shader cores
- 384-bit GDDR5 memory interface, 6GB capacity
- GeForce GTX480: Graphics Enthusiast
  - Full Microsoft DirectX 11 Shader Model 5.0
  - 32x coverage sample antialiasing (AA)
  - 128-bit floating point high dynamic-range lighting with AA
- Tesla C2070: Compute HPC
  - CUDA programming: C, C++, OpenCL, DirectCompute, or Fortran
  - 515 GigaFLOPS double-precision peak floating point
  - ECC Register Files, L1/L2 caches, shared memory and DRAM

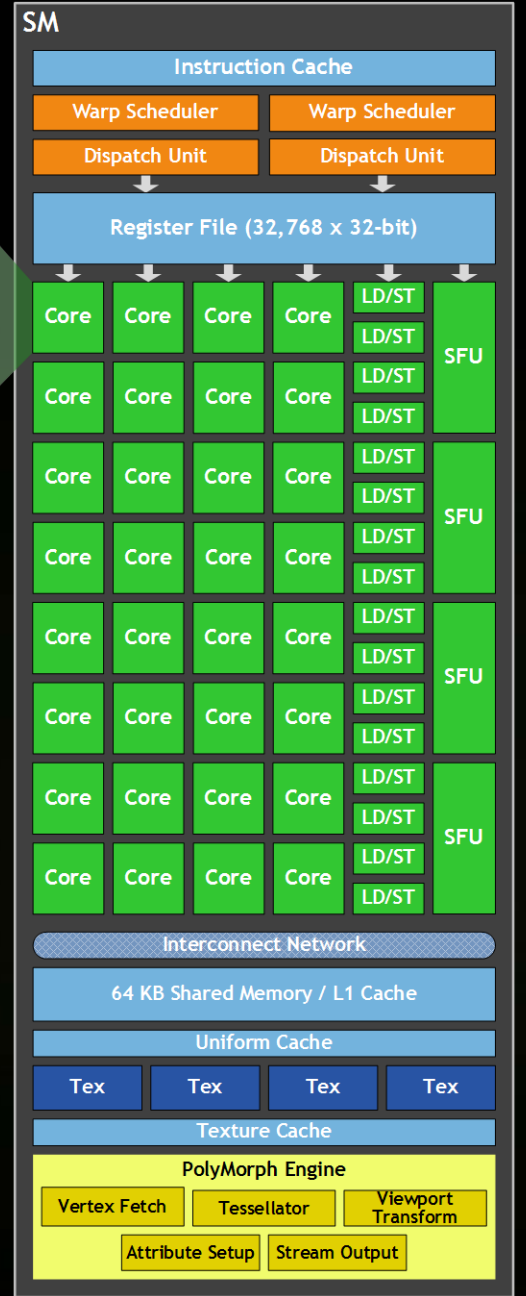
GF100 Architecture

- 512 CUDA cores
- 16 PolyMorph Engines
- 4 raster units
- 64 texture units
- 48 ROP units
- 384-bit GDDR5

GF100 SM Architecture

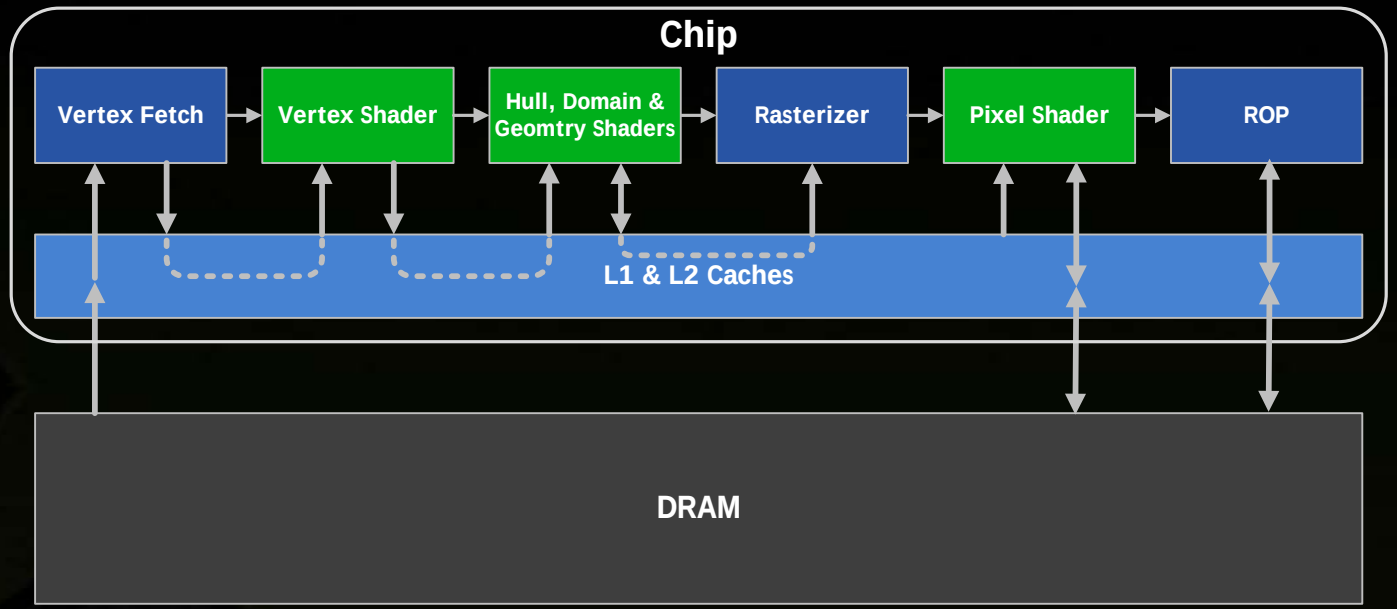
- SM – Streaming Multiprocessor
- 32 CUDA cores: 4x GT200
  - GT200 – previous high-end GPU by NVIDIA
- 48 or 16KB of shared memory: 3x GT200
- 16 or 48KB of L1 cache: No L1 on GT200
- ISA improvements
  - 32-bit integer operations
  - FMA (fused multiply add) IEEE-754 2008
- 4 Texture units
- 1 PolyMorph Engine





Cache Architecture for Graphics

- Data stays on die
- L1 cache
  - Register spilling
  - Stack ops
  - Global LD/ST
- L2 Cache
  - Vertex, SM, Texture and ROP Data



Cache Architecture Comparison

	GT200	GF100	Benefit
L1 Texture Cache (per SM)	12 KB	12 KB	Fast texture filtering
Dedicated L1 LD/ST Cache	✗	16 or 48 KB	Efficient physics and ray tracing
Total Shared Memory	16KB	16 or 48 KB	More data reuse among threads
L2 Cache	256KB (TEX read only)	768 KB (all clients read/write)	Greater texture coverage, robust compute performance

DX10 Game Problems, too little detail

- Geometry throughput is challenge
  - Flat head
- Silhouette results from coarse triangulation
- There are other artifacts

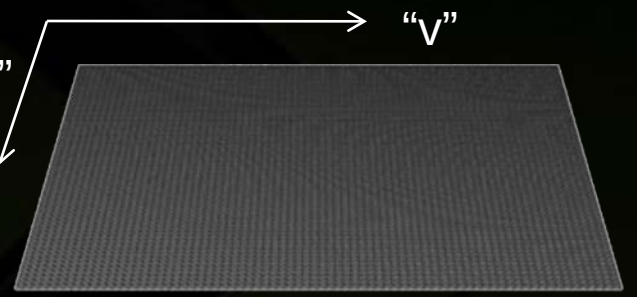
Image from Far Cry® 2, courtesy of Ubisoft

Offline Film Rendering uses Tessellation

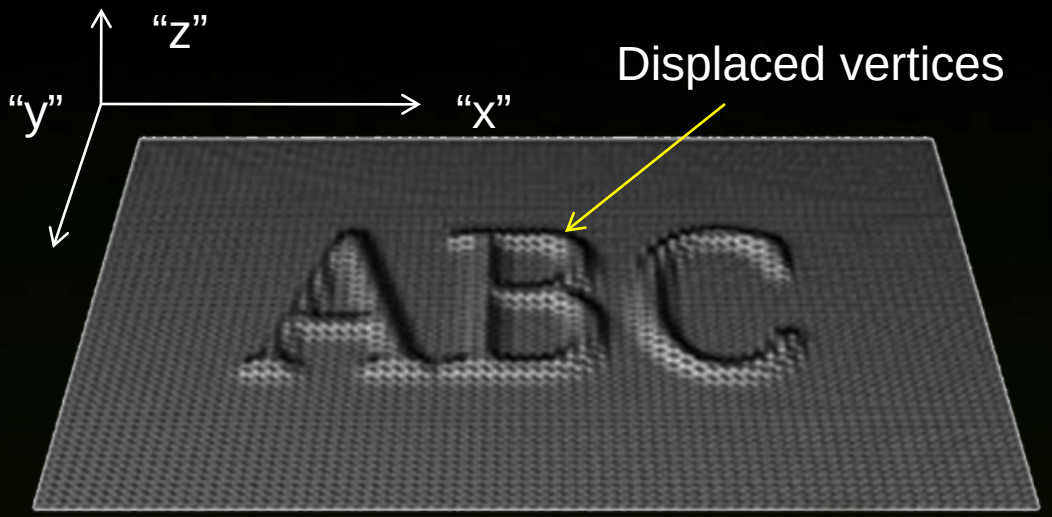
- Tessellation + displacement mapping modelling standard
- Rich geometric detail
- More Geometry, Takes More time

Tessellation & Displacement Maps
Add Geometric Detail

Original Geometry



Tessellated Geometry



Final Geometry

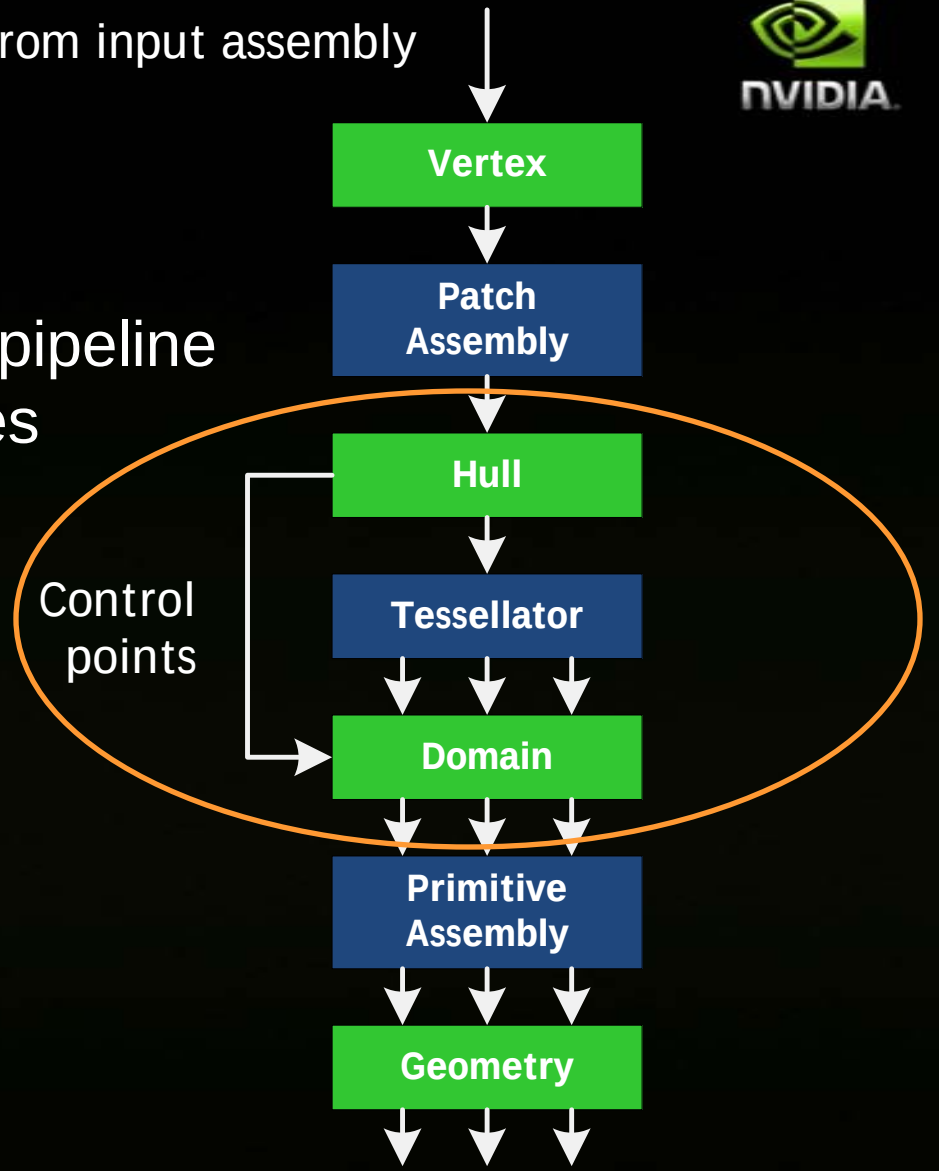
Displacement Map

http://en.wikipedia.org/wiki/File:Displacement.jpg

Tessellation in DirectX 11

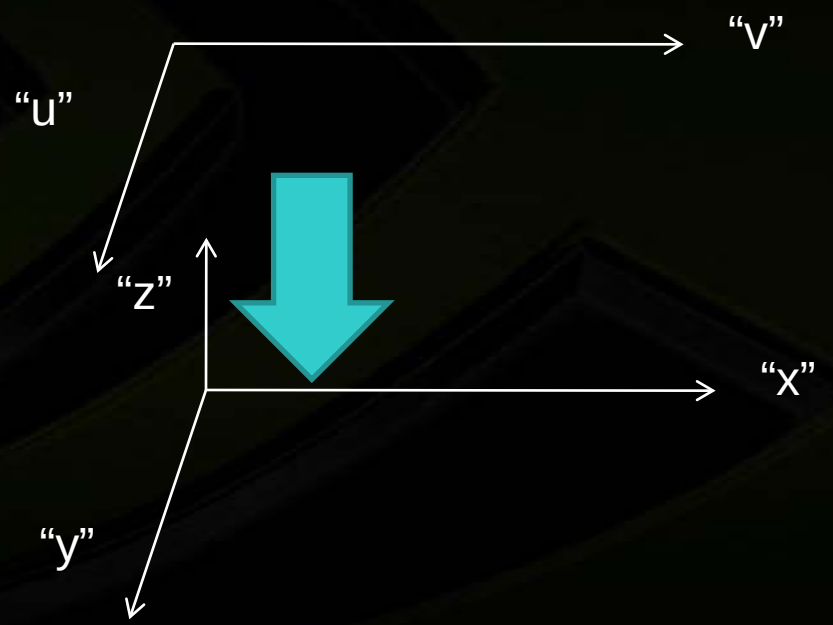
- **Hull shader**
  - input new patch primitive
  - Outputs tessFactors, modes
  - Runs pre-expansion (1 arrow)
- **Fixed function tessellation stage**
  - input TessFactors and modes
  - Outputs triangles and lines
  - On chip data expansion (3 arrows)

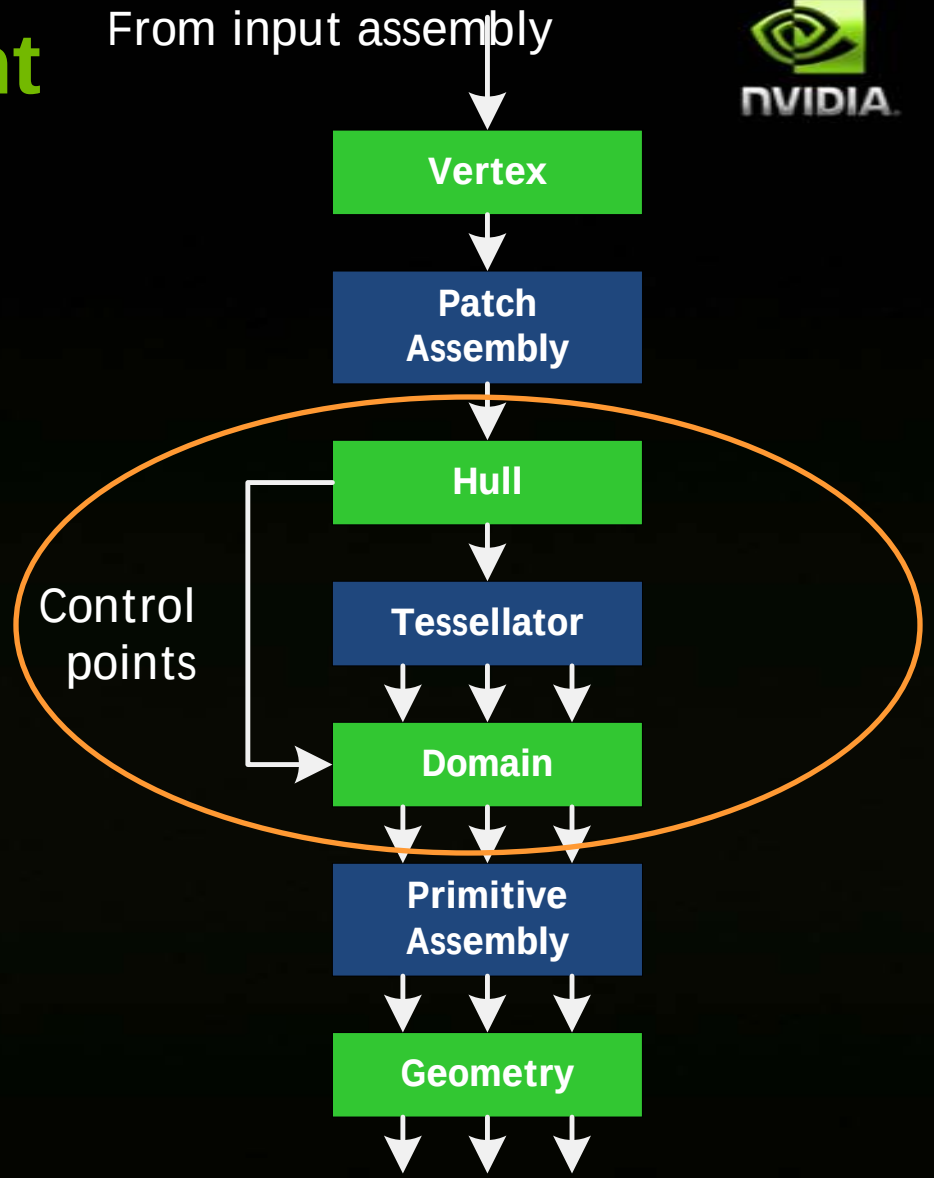
New pipeline stages



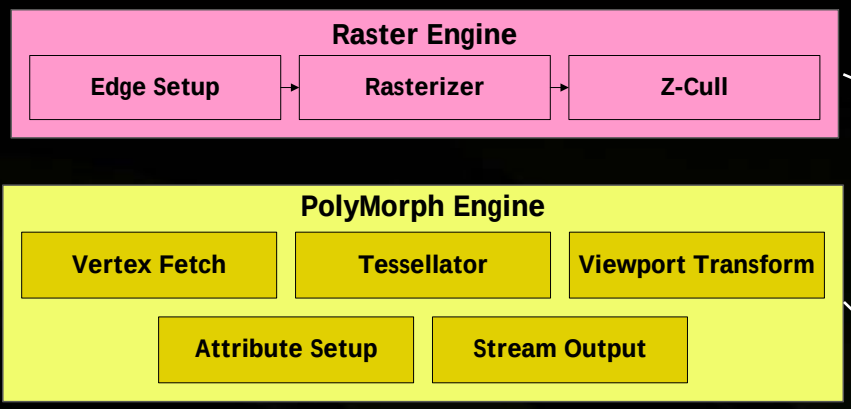
Tessellation in DirectX 11, cont

- Domain shader
  - Input TessFactors, (u,v) domain points
  - Outputs vertices in (x,y,z,w)





GF100 Scalable Parallel Implementation



Distributed, parallel geometry
4 Raster Engines
16 PolyMorph Engines

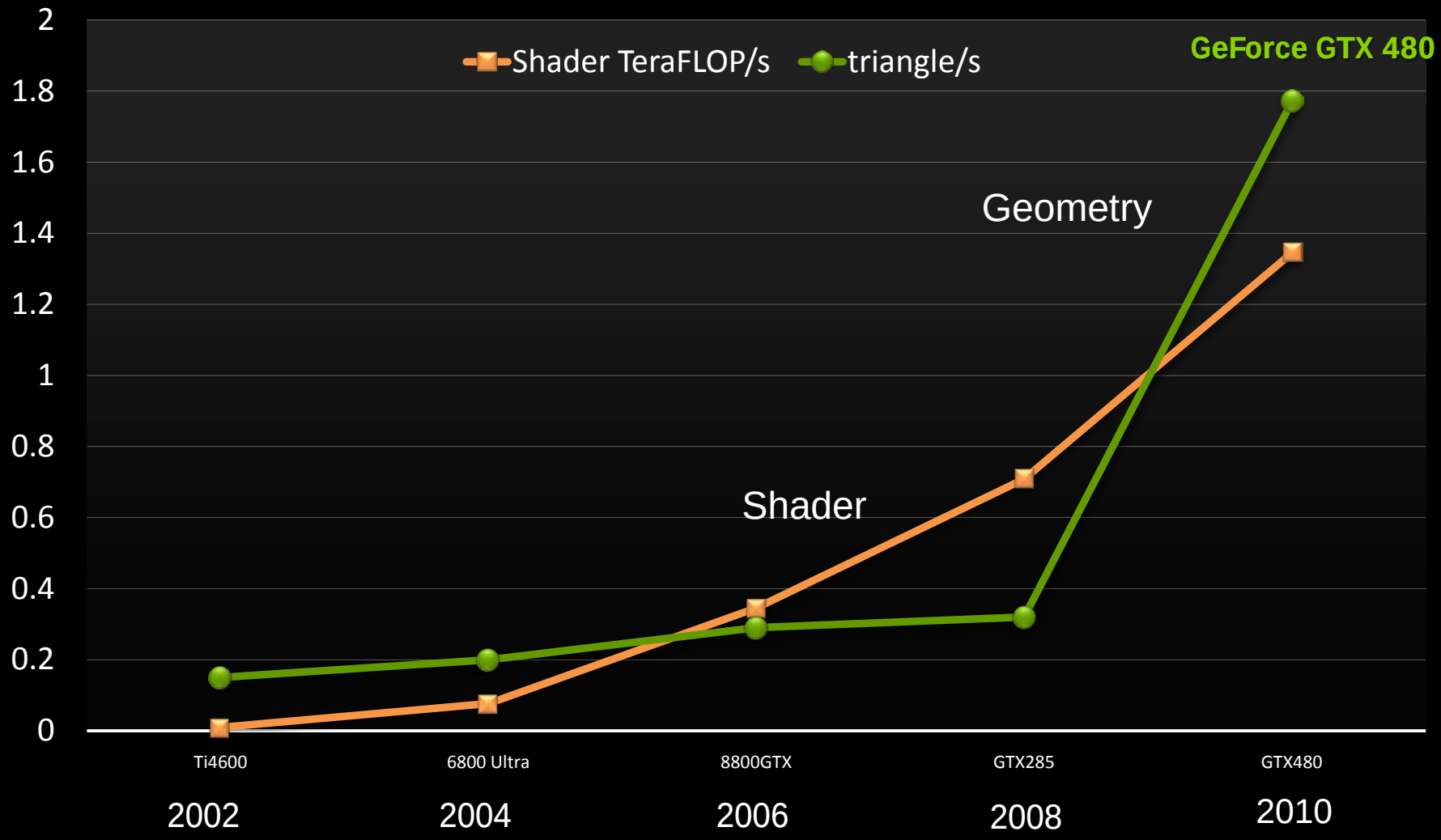
8x geometry performance vs. GT200

Enables > 2.5 triangles/clock

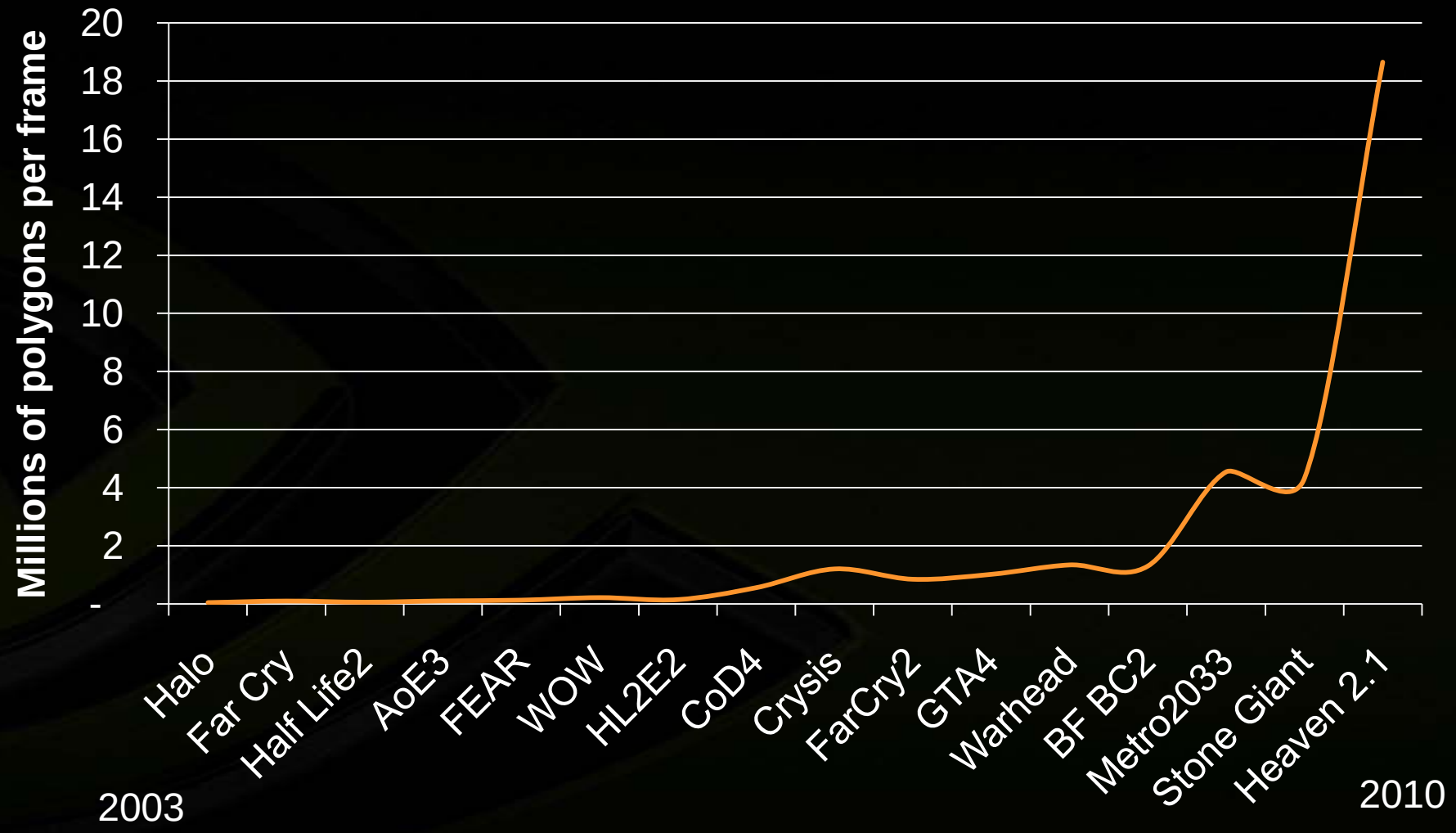
GF100 Enables Geometric Realism

- Tessellation provides interactive smooth silhouette
  - Film like capabilities in games
- More geometry detail—3D Stereo viewing benefits
  - Dynamic shadows
  - 3D Vision™
- Memory footprint & BW savings
  - Store coarse geometry, expand on-demand
  - Enables more complex animations
- Scalability
  - Dynamic LOD allows for performance/quality tradeoffs
  - Scale into the future – resolution, compute power

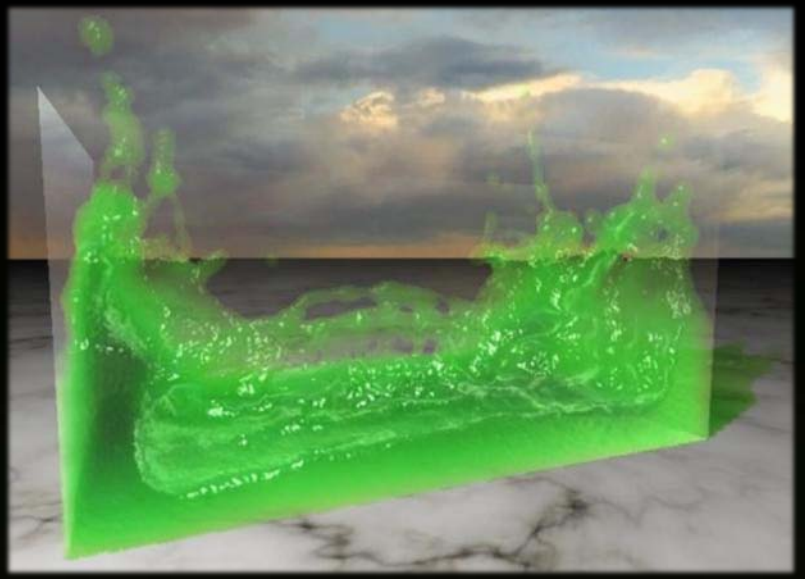
Dramatic Increase in Geometry Capability



Advance in Geometric Complexity

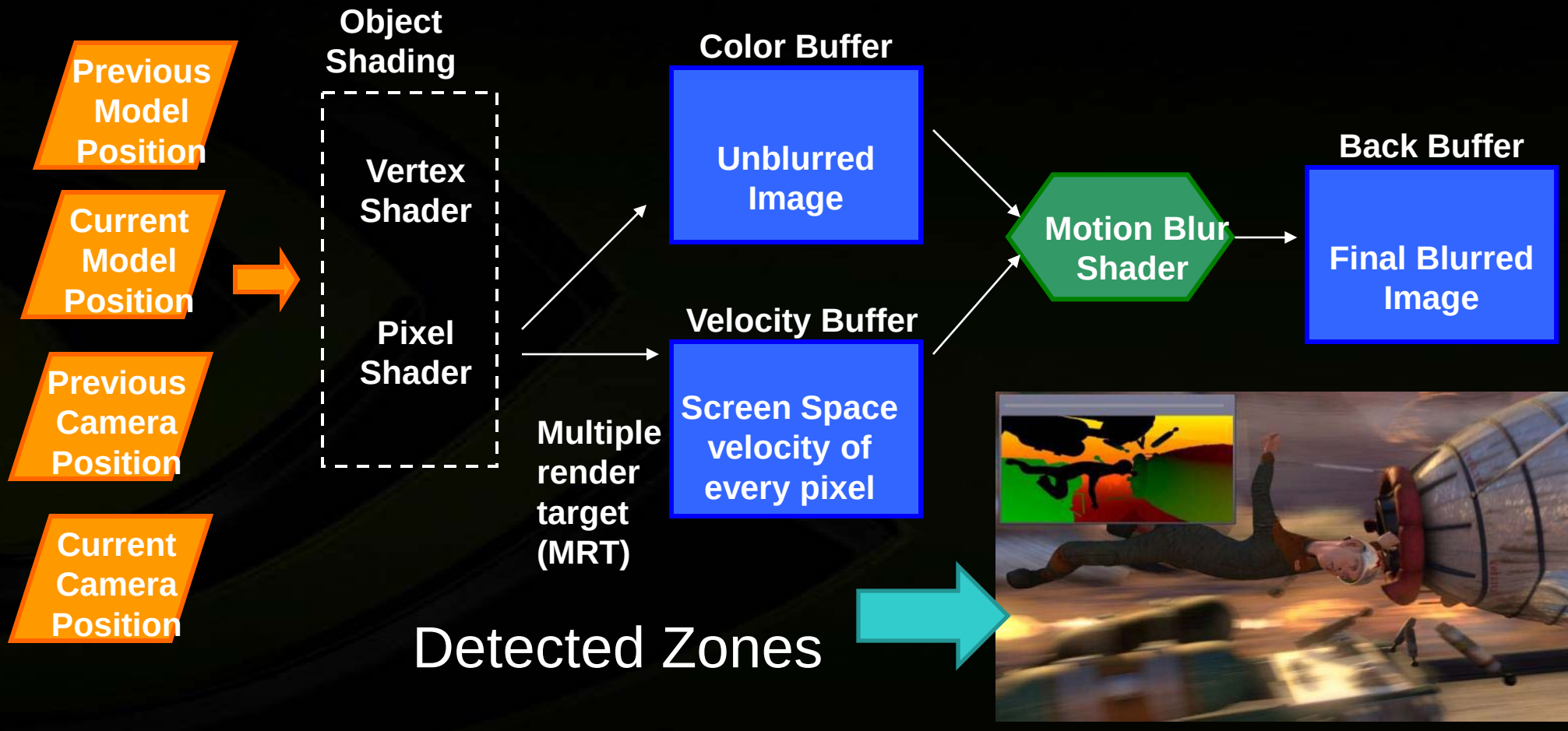


Physics: PhysX® Example: Fluid Simulation



- PhysX® is NVIDIA physics engine
  - Uses CUDA SW stack
  - Used in over 150 computer games
- Particle-based fluid simulation (SPH)
  - 128,000 particles
  - Includes surface tension
- 3x speed up going from GT200 to GF100
  - 67 to 200 fps
- Games fluid for water splashes, mud, blood…

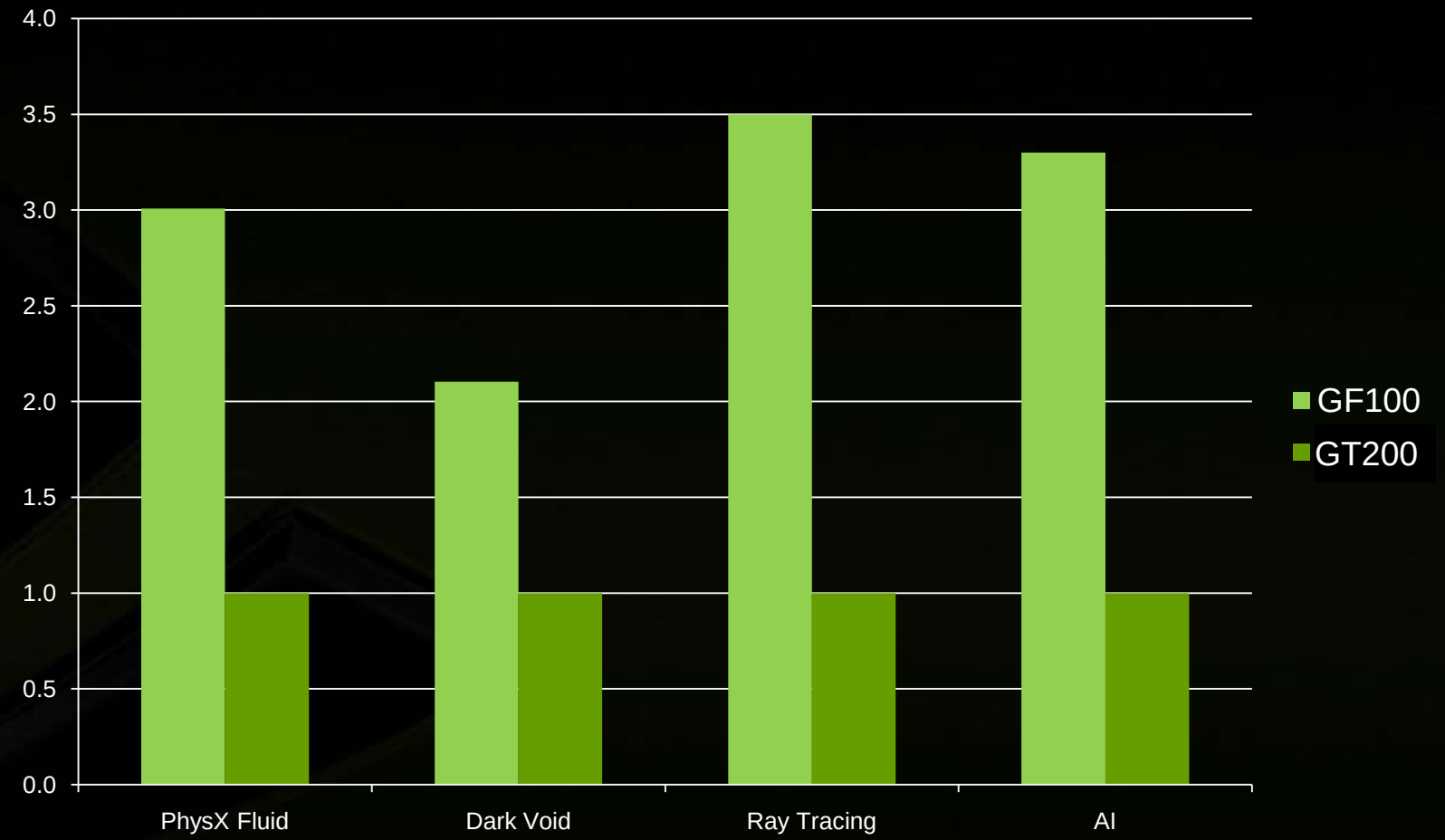
Computational Graphics: Image Processing – Motion Blur



Computational Graphics: Ray Tracing

- Combine Rasterization with Raytracing
  - Rasterize primary rays
  - Raytrace shadows and reflections
- 4x faster than GT200
  - Efficient use of cache architecture

GF100 GPU Compute Performance

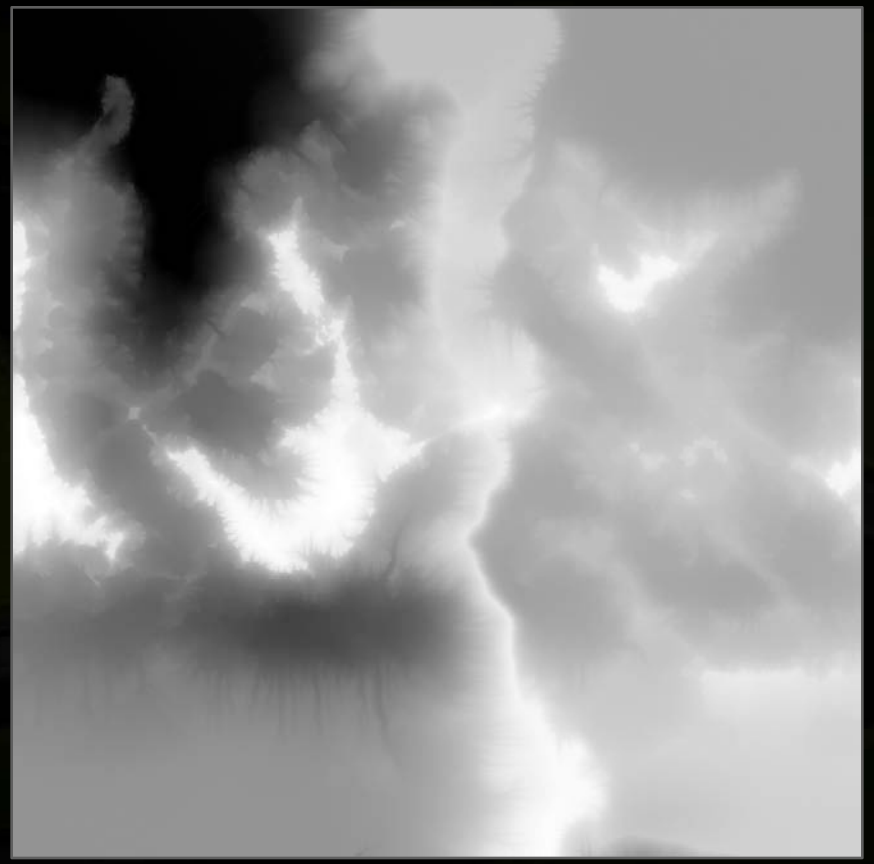


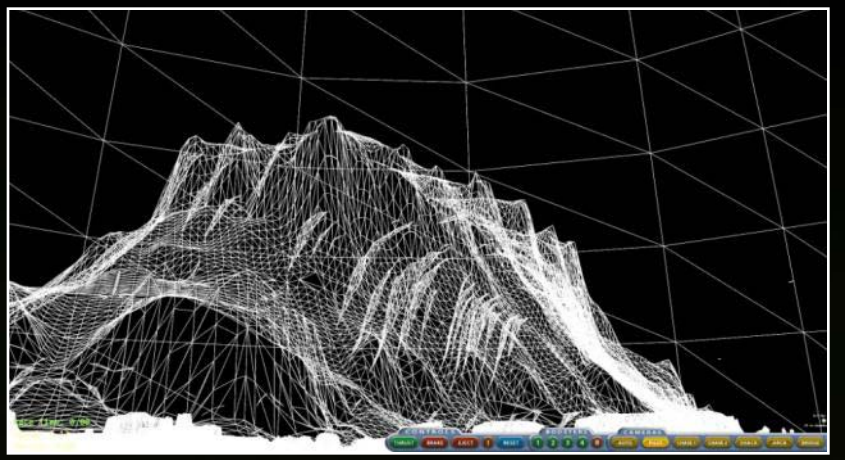
Demo: SuperSonic Sled--tessellation, physics, and computational graphics

SuperSonic Sled: Tessellation - Terrain

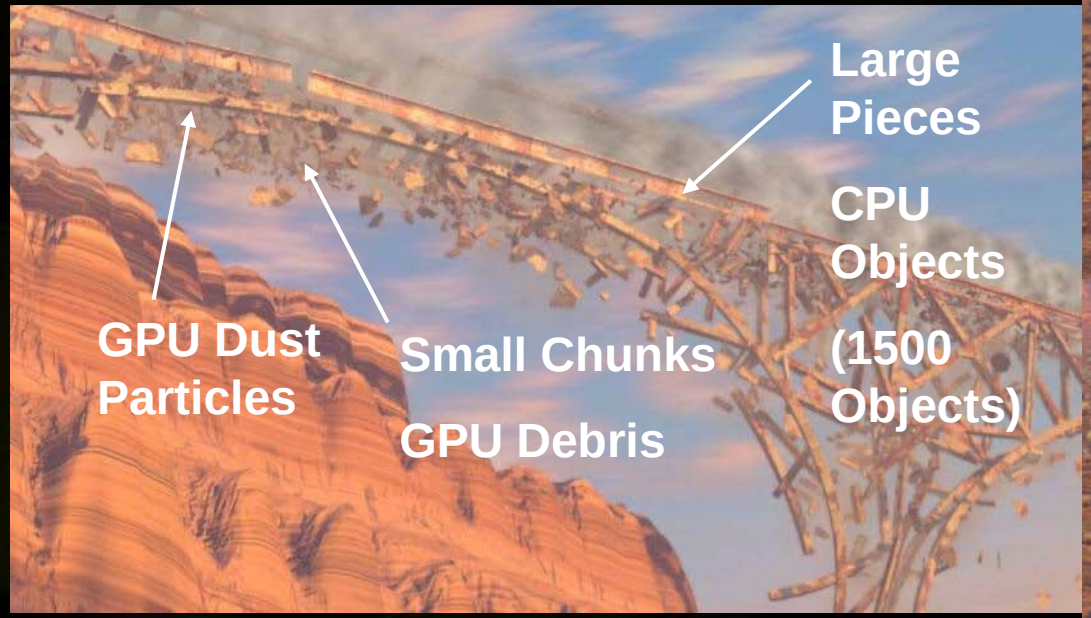
NVIDIA

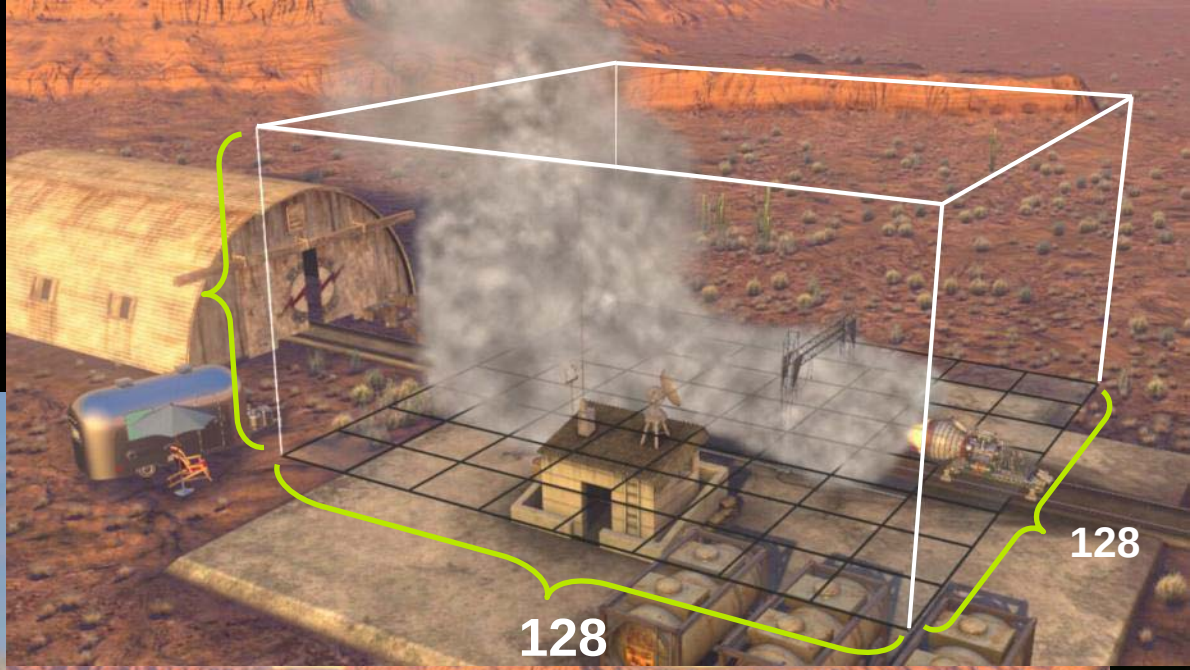
Terrain Displacement Map





SuperSonic Sled: Physics – Smoke, Bridge, Fireballs





SuperSonic Sled: Computational Graphics-Motion Blur

Demo

Conclusions

- GF100 Improves Geometry processing up to 8X over GT200
- New Architecture and New family of chips
- GF100 Architecture breaks 1 triangle per clock barrier
  - More than 2.5 Triangles/clock
  - High throughput tessellation has benefits for interactive gaming
- Demonstration of
  - Tessellation
  - Physics
  - Computational Graphics